



zen um einen Tisch herum, auf dem das Bild möglichst rasch zusammengesetzt werden soll. Jeder darf Teile auf dem Tisch anlegen, wenn er meint, eine passende Stelle gefunden zu haben – oder wieder welche wegnehmen, die er falsch angelegt hatte. Auf diese Weise können die Mitspieler zusammenwirken in der Herstellung des Bildes, ohne direkt zu kommunizieren. Sie reagieren auf Änderungen der Konstellation auf dem Tisch, indem sie selbst solche Änderungen durchführen.

### Runde um Runde

Eine Stufe weiter im Bild: Nehmen wir an, diese Spielregeln führen zu allzu großem Durcheinander beim Spielverlauf, weil immer mehrere gleichzeitig Teile legen oder wegnehmen wollen. Es wird beschlossen, daß immer nur jeweils einer das teilweise fertiggestellte Puzzlebild verändern darf. Die Entscheidung, wer bei mehreren Bewerbern in einer Runde am Bild arbeiten darf, wird einem neu ernannten Spielleiter übertragen. Der soll in jeder Runde entscheiden, welcher der Bewerber den nützlichsten Beitrag zur Fertigstellung des Bildes leistet und ihm den Zuschlag erteilen.

Damit haben wir schon die drei wesentlichen Bestandteile von Blackboard-Systemen: Das Blackboard selbst als globale gemeinsame Datenstruktur (der Tisch mit dem Puzzlebild), unabhängige Spezialisten, 'Wissensquellen' genannt, die je nach Zustand Veränderungen am Blackboard vornehmen, und eine Kontrollinstanz, die den Zugriff auf das Blackboard regelt (der Spielleiter). Alle wirken zusammen bei der Lösung, aber kommunizieren nicht direkt miteinander.

In der Puzzleanalogie sind wichtige Eigenschaften eines Blackboard-Systems zu sehen:

**Opportunistisches Vorgehen:** Die Wissensquellen werden dynamisch koordiniert. Es gibt keine festgelegte Reihenfolge der Aufrufe. In jeder Runde wird neu gefragt, welche Beiträge die Spezialisten leisten können, und der nützlichste von ihnen wird ausgesucht.

**Unabhängigkeit der Wissensquellen:** Sie interagieren nicht, sind also nur sehr lose gekop-

## Tafelgeschäfte

**Blackboard-Systeme meistern Komplexität**

**Thomas J. Schult**

Stellen Sie sich das Gewimmel an der Börse vor. Vorne eine riesige Tafel, auf der die Kurse verzeichnet sind. Anleger kommen, deren Käufe und Verkäufe die Kurse verändern. Die veränderten Kurswerte auf der Tafel veranlassen wiederum andere, zu kaufen oder zu verkaufen.

Was an der Börse keinem übergeordneten Ziel zu dienen scheint, mag als Metapher die Entstehung einer Software-Architektur beflügelt haben, die in der Künstlichen Intelligenz (KI) oft die Methode der Wahl wurde, wenn komplexe, inhomogene und schlecht strukturierte Bereiche zu bearbeiten waren: Blackboard-Systeme. Vorgeschlagen wurden sie vor 30 Jahren vom kürzlich verstorbenen Kognitionswissenschaftler Allen Newell; die erste größere Anwendung fanden sie vor 20 Jahren in HEARSAY, einem der ersten Systeme, die kontinuierlich gesprochene

Sprache verstehen konnten. Früher waren Blackboard-Systeme auf die Dinosaurier der KI abonniert: LISP-Maschinen, sündhaft teure Spezialhardware. Deren Zeit ist abgelaufen, und inzwischen ist mit GBB sogar das erste Entwicklungssystem auf PC-Basis erhältlich. Grund genug, einen Blick auf die Prinzipien dieser Systemarchitektur zu werfen, bevor wir uns in einem der nächsten Hefte mit GBB befassen.

Eine weitere Analogie: Nehmen wir ein Puzzle, dessen Teile gleichmäßig auf eine Gruppe von erfahrenen Puzzlelegern verteilt werden. Sie sit-

pelt über ihren Zugriff auf einen gemeinsamen Speicher. Das ermöglicht größtmögliche Flexibilität bei der Gestaltung der Wissensquellen. Wie sie Wissen repräsentieren und wie sie es verarbeiten, ist ihre Sache.

Ein Modul kann ein Regelsystem sein, ein anderes ein neuronales Netz, ein drittes eine C-Prozedur. Die einzige Bedingung ist, daß die Wissensquellen die für sie relevanten Einträge auf dem Blackboard lesen können.

**Aktivierung der Wissensquellen** durch Veränderungen des Blackboards: Normalerweise lösen Operationen auf dem Blackboard (Schreiben, Ändern, Löschen) Aktivierungen einzelner Wissensquellen aus. Bei einer Aktivierung geben die Wissensquellen zu erkennen, daß sie einen Beitrag zur Problemlösung auf dem Blackboard leisten können.

**Kontrolle durch Auswahl aktivierter Wissensquellen:** Da oft ein Blackboard-Zustand mehrere Wissensquellen aktiviert, muß der Zugang zum Blackboard durch ein Kontrollmodul geregelt werden. Es kann etwa eine Abschätzung von Kosten und Qualität eines Beitrags durchführen. Die in diesem Sinne beste aktive Wissensquelle darf auf dem Blackboard operieren. Die anderen müssen warten, haben aber in der nächsten Runde wieder eine Chance, sofern der neue Blackboard-Zustand sie weiterhin aktiviert.

**Inkrementelle Problemlösung:** Keiner der in den Wissensquellen realisierten Verfahren reicht aus, um die Lösung zu finden. Sie wird nach und nach konstruiert und entwickelt sich auf dem Blackboard.

### Rahmen liefern

Herkömmliche Methoden der Software-Konstruktion sehen ganz anders aus. Da wird eine Spezifikation des Programms erstellt, das Problem anhand einer funktionalen Hierarchie in einfachere Teilmodule zerlegt. Dabei wird Wert auf eine mehrfache Nutzung einzelner Module gelegt und die Kontrolle etwa durch feste Aufrufsequenzen definiert.

Leider sind gerade in der KI nicht alle Aufgaben von vornherein so exakt spezifizierbar. Um komplexe Problemstellungen

|                              | Regelbasiertes Expertensystem  | Fallbasiertes Expertensystem                                     | Objektorientiertes System   | Blackboard-System  |
|------------------------------|--|--|---|--|
| Konzeptuelles Modell         | Zielreduktion<br>Problemlösen als Beweisen   | Problemlösen durch Analogie<br>Lernen durch Zuwachs an Erfahrung | Gliederung nach Objekten, nicht Funktionen<br>Vererbung von Eigenschaften und Methoden<br>Direkte Kommunikation | Opportunistische, inkrementelle Problemlösung<br>Lose Kopplung unterschiedlicher Repräsentations- und Verarbeitungsmittel<br>Indirekte Kommunikation |
| Organisation der Komponenten | Zielhierarchie<br>Wenn ... dann ... - Regeln   | "Gedächtnis" mit früheren Fällen<br>Anpassungsheuristiken        | Klassenhierarchie   | Globaler Mehrebenen-speicher<br>Modulare Spezialisten  |
| Kontrolle                    | Regelverkettung gemäß Inferenzmechanismus  | Ähnlichkeitsmaß bestimmt Auswahl früherer Fälle                  | Kontrollübertragung durch Nachrichten   | Spezialisten als Dämonen<br>Explizite Kontrollstrategie<br>Agenda aktivierter Spezialisten   |
| Entwicklungswerkzeuge        | Shells (ART-IM, KEE, Kappa, Nexpert Object, TWAICE,...)<br>Regelsprachen (OPS, PROLOG,...) | Shells (ART-IM, ESTEEM, ReMind,...)                              | Objektorientierte Sprachen (Simula, Smalltalk, CommonLisp mit CLOS, C++,...)                                    | Shells (GBB, BB1, HEARSAY-III, AGE,...)  |

### KI ist mehr als Regeln: Architekturen für wissensbasierte Systeme.

gen handhaben zu können, kommt man oft um 'explorative' Programmieren nicht herum, muß also die Vermischung von Programmdesign (Spezifikation) und Implementierung in Kauf nehmen. Da ist ein konzeptueller Rahmen für die Organisation verschiedener Wissensarten, Problemlöse- und Kontrollschritte hilfreich. Bestimmte Architekturen von KI-Software haben sich als hinreichend allgemein herauskristallisiert, um einen solchen Rahmen zu bieten: klassische regelbasierte oder fallbasierte Expertensysteme, objektorientierte Systeme oder eben Blackboard-Systeme. Sie stellen jeweils eine eigene Weise der Organisation und Kontrolle von Bereichswissen und Systemkomponenten bereit.

### Unabhängige Entscheidungsträger

Bei Blackboard-Systemen lassen sich Aufgaben auf eine andere Weise zerlegen als bei der klassischen strukturierten Software-Entwicklung: Gesucht werden nicht hierarchisch abhängige Module und gemeinsame Teilprobleme, sondern funktional möglichst unabhängige Einheiten, damit keine Interaktion zwischen ihnen nötig ist. Kontrollinformationen verwaltet das System separat. Durch diese Trennung in unabhängige Spezialisten erreicht man eine maximale Flexibilität – sowohl hinsichtlich der Kontrollstrategie

als auch hinsichtlich der in den Wissensquellen verwendeten Verfahren. Auch die Aufteilung der Aufgabe auf mehrere Entwickler wird durch die Blackboard-Architektur erleichtert, da nur wenige Schnittstellen bestehen.

Regelbasierte Systeme weisen ebenfalls eine starke Modularität auf; darüber hinaus muß wie bei Blackboard-Systemen in jedem Zyklus eine Entscheidung getroffen werden, welche der aktivierten Module (sprich: Regeln) zuzuführen ist. Jedoch handelt es sich um eine Modularisierung auf einer viel feineren Auflösungsstufe. Zudem weisen regelbasierte Systeme im Gegensatz zu Blackboard-Systemen einen festen Inferenzmechanismus und ein festes Repräsentationsformat für die Module auf.

Zurück zur Beschreibung der Blackboard-Architektur: Das Blackboard als gemeinsamer Datenspeicher ist das Herzstück des Systems. Hier werden Zwischenschritte und Resultate dokumentiert. Die Struktur und Sprache der Blackboard-Einträge definiert den Raum möglicher Lösungen. In der Regel findet sich bei Blackboard-Systemen eine hierarchische Strukturierung des Blackboard, die entweder nach Abstraktionsstufen (wie bei HEARSAY-II) oder nach einer Teil-Ganzes-Beziehung geordnet ist. Weitere Strukturmerkmale ergeben sich je nach Anwendungsbereich – oft ist eine zeitliche Di-

mension zu berücksichtigen, die eine zusätzliche Strukturierung nach Zeitintervallen nahelegt. In einigen Anwendungen haben sich auch mehrere Blackboards als sinnvoll erwiesen.

### Duales System

Die Wissensquellen spielen in einem Blackboard-System die Rolle von unabhängigen Spezialisten für bestimmte Aufgaben. Sie operieren in der Regel auf dem Blackboard und legen dort ihre Resultate ab. In der Regel transformieren sie Blackboard-Daten von einer Hierarchiestufe auf eine andere. Sie bestehen oft aus Prozeduren oder Regelmengen. In manchen Anwendungen können sie auch direkt externe Signale verarbeiten und gegebenenfalls interpretiert auf dem Blackboard ablegen. Direkte Kommunikation mit anderen Wissensquellen ist allerdings im allgemeinen verboten.

Wissensquellen sind zweigeteilt. Ein (Bedingungs-)Teilsystem ist dafür zuständig, diejenigen Blackboard-Zustände zu charakterisieren, die einen Beitrag der Wissensquelle als sinnvoll erscheinen lassen. Das zweite (Aktions-)Teilsystem liefert dann den tatsächlichen Beitrag, wenn er gewünscht wird. Um nicht bei jeder Änderung eines Blackboard-Eintrags alle Bedingungs-Teilsysteme von Wissensquellen durchtesten zu müssen, können Klassen von Blackboard-Änderungen mit

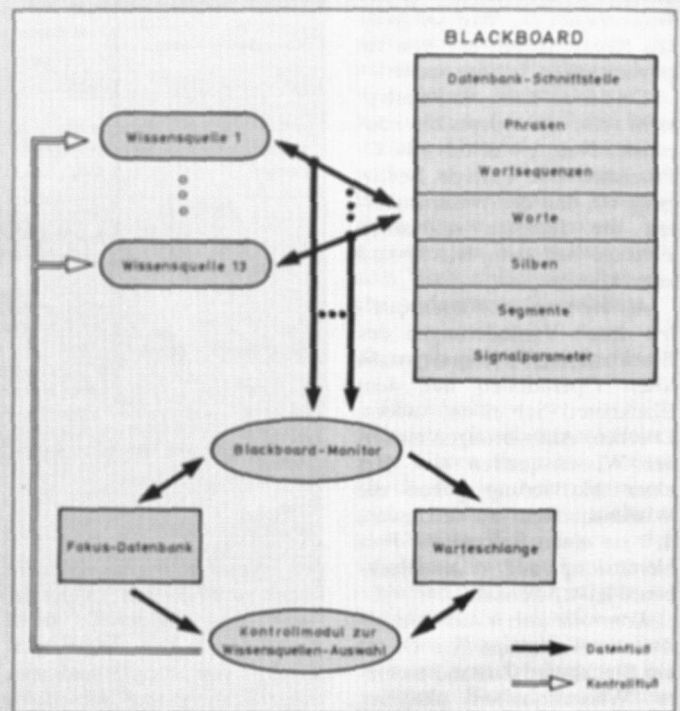


## Spracherkennung mit HEARSAY-II

HEARSAY-II erregte Aufsehen als System zum Verstehen kontinuierlich gesprochener Sprache, aber sein wichtigster Beitrag zur KI lag in der Art und Weise, wie dies erreicht wurde. Die in HEARSAY erstmals verwendete Blackboard-Architektur wurde zu einem klassischen konzeptuellen Modell für die Organisation von komplexen intelligenten Systemen.

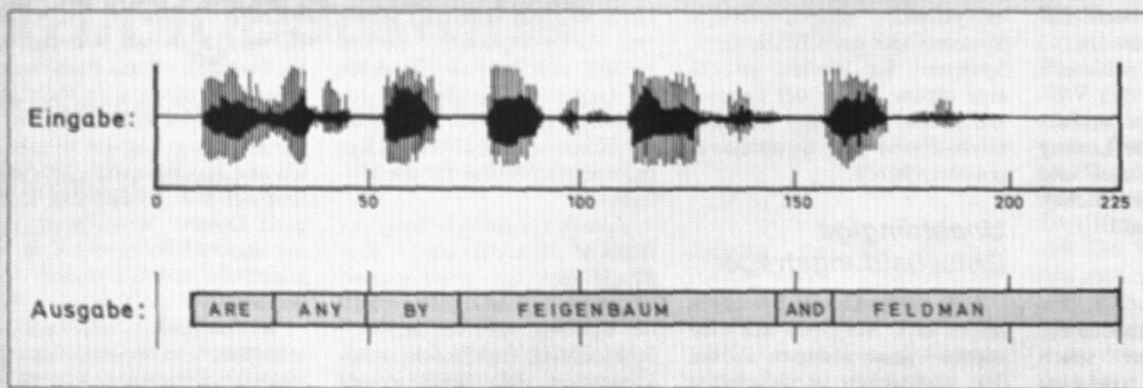
HEARSAY-II wurde als Oberfläche für eine KI-Literaturdatenbank entwickelt. Es verarbeitet das akustische Signal eines Benutzers, der eine Literaturanfrage stellt. HEARSAY-II kennt etwa 1000 Wörter. Neben den Mehrdeutigkeiten gesprochener Sprache wird die Aufgabe noch durch verrauschte Daten und Eigenheiten der Sprecher (Vokabular, Aussprache, Syntax) erschwert. Nach einer gewissen Trainingszeit mit einem Sprecher erreicht HEARSAY-II eine zu 90 % korrekte Erkennung der Sprache in Form einer geschlossenen semantischen Interpretation und kann diese anschließend in eine Datenbankabfrage umsetzen. Zwischen Sprechsignal und Datenbank-Interface liegen fünf unterschiedliche Stufen, nach denen die Daten auf dem Blackboard organisiert sind: Segmente (kleinste Lauteinheiten), Silben, Wörter, Wortketten und Phrasen. Neben dieser Einteilung, die man als Abstraktionshierarchie bezeichnen kann, ist das Blackboard nach Zeitintervallen strukturiert; außerdem können jeweils mehrere Alternativen verzeichnet werden. Ein Eintrag auf dem Blackboard ist also eine (alternative) Hypothese für eine bestimmte Abstraktionsstufe einer Äußerung in einem bestimmten Zeitintervall.

Die meisten Wissensquellen überführen Blackboard-Einträge von einer niederen in eine höhere Abstraktionsstufe, zum Beispiel schlägt MOW ausgehend von Hypothesen über erkannte Silben Wörter vor, die diese Silben in der richtigen Reihenfolge enthalten. Es gibt aber auch Wissensquellen, die anders gerichtet sind. So bestimmt PREDICT Worte, die eine gegebene Phrase unter Beachtung der Syntax umschließen können. (Die Syntax von HEARSAY-II ist so weit an den Bereich angepaßt, daß es für ein nachfolgendes Wort nur durchschnittlich 17 Kandidaten aus dem Wortschatz gibt.) Wieder andere Wissensquellen bleiben auf einer Ebene – etwa RPOL, die Hypothesen auf ihre Glaubwürdigkeit



**Spezialisten schauen auf die Tafel: HEARSAY-II etablierte die typische Blackboard-Architektur.**

prüft. Auch hinsichtlich der Zeitachse können Wissensquellen unterschiedlich gerichtet sein: Sowohl ein Schluß von einem Ereignis auf ein nachfolgendes als auch auf ein gleichzeitiges oder vorhergehendes ist möglich. Alle Blackboard-Einträge werden als vorläufig behandelt und können später wieder revidiert werden.



**Keine Fieberkurve, sondern gesprochene Sprache: Aber wie wird daraus ein Satz fürs Programm?**

möglicherweise anwendbaren Wissensquellen assoziiert werden, beispielsweise in einem separaten Index, der bei der Entwicklung mit zu spezifizieren ist.

Neben solchen Wissensquellen, die die Lösung auf dem Blackboard direkt vorantreiben, kann es auch solche geben, die nur Kontrollzwecken dienen

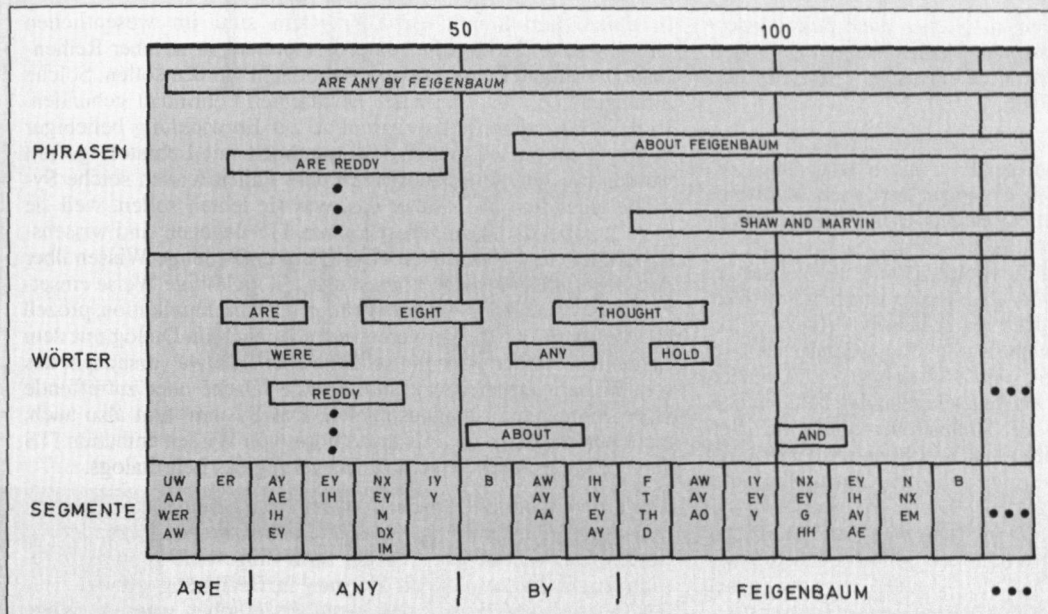
und Informationen bereitstellen, die das Kontrollmodul für die Auswahl von Beiträgen anderer Wissensquellen nutzt.

Nicht jede Änderung auf dem Blackboard muß gleich Tests auf anwendbare Wissensquellen nach sich ziehen – ein Blackboard-Monitor kann zwischengeschaltet sein, der nur bestimmte Änderungen als aus-

lösende Ereignisse klassifiziert und an die Wissensquellen weitergibt.

Im Gegensatz zu Standard-Entwicklungsverfahren für Software wird die Kontrolle in Blackboard-Systemen als separate, dynamisch zu bearbeitende Aufgabe aufgefaßt und ist nicht in den Problemlösemodulen fest verdrahtet. Das Kontrollmodul

verwendet oft einen expliziten 'Aufmerksamkeitsfokus', der die bei der Problemlösung entstehenden Kontrollinformationen bündelt. Im Fokus steht entweder ein Teil des Blackboards in Form von Zwischenschritten beziehungsweise Teillösungen, die zunächst weiterverarbeitet werden sollen, oder Wissensquellen, die bevorzugt anzu-



**Keiner kann alles:  
Spezialisten  
transformieren  
Blackboard-Einträge.**

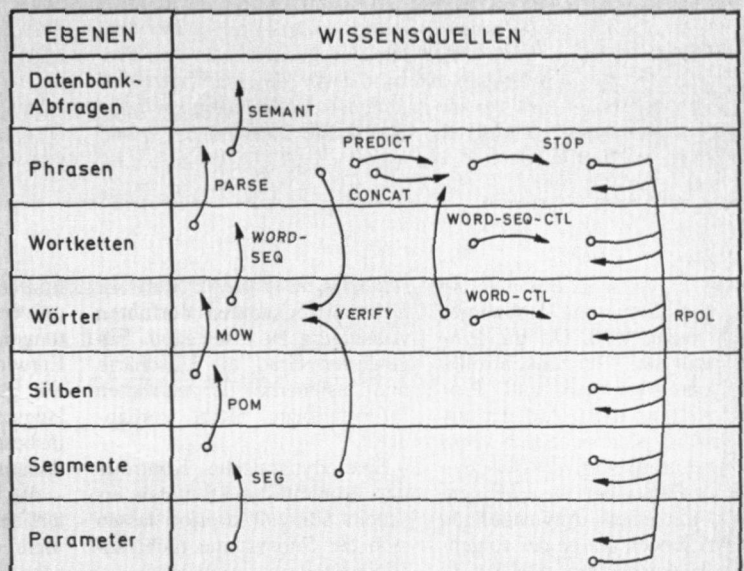
eine Aktivierung des entsprechenden Aktionsteils der Wissensquelle erzeugt und in die Warteschlange für eine mögliche spätere Ausführung eingereiht.

Verschiedene Faktoren steuern die Auswahl eines aktivierten Bedingungs- oder Aktionsteils. Zunächst werden die Kosten für die Ausführung abgeschätzt, außerdem die Zu-

verlässigkeit der Blackboard-Einträge berücksichtigt, die die Aktivierung auslösten. Schließlich wird die Zuverlässigkeit der Wissensquellen selber in Betracht gezogen, ferner geprüft, ob sie solche Elemente auf dem Blackboard weiterverarbeiten können, denen bereits eine hohe Glaubwürdigkeit zugesprochen wird.

Die HEARSAY-II-Wissensquellen haben einen Bedingungsteil, der sowohl die Blackboard-Situation charakterisiert, in der sie anwendbar ist, als auch einen Ausblick auf den Beitrag der Wissensquellen gibt, und einen Aktionsteil, der den Beitrag in Form von Änderungen auf dem Blackboard tatsächlich durchführen kann. Beide Teile sind komplexe Programme, deren Aufrufe vom Kontrollmodul separat verwaltet werden. Schon ein kurzes zusätzliches Störgeräusch in der eingegebenen sprachlichen Äußerung kann zu einer veränderten Aufrufsequenz der Wissensquellen führen.

Nimmt man auf dem Blackboard eine Änderung vor, so prüft der Blackboard-Monitor anhand der Einträge in der Fokus-Datenbank, welche Wissensquellen potentielle Kandidaten für eine Aktivierung abgeben. In der Fokus-Datenbank steht beispielsweise, daß bei Hinzufügung eines Segmentes auf dem Blackboard die Wissensquellen POM und VERIFY sich möglicherweise aktiviert lassen. Die Bedingungsteile der beiden Wissensquellen werden anschließend in die Warteschlange aufgenommen. Die am nützlichsten erscheinende Aktivierung kommt zur Auswahl und der entsprechende Bedingungsteil mit dem aktuellen Blackboard-Zustand zur Auswertung. Trifft die Bedingung zu, so wird



**Survival of the fittest: Die meisten Blackboard-Daten werden wieder verworfen.**

wenden sind (oder beides). Aufgabe des Kontrollmoduls ist es dann, eine geeignete Wissensquelle und sie aktivierende Blackboard-Daten zu finden, die den Fokus berücksichtigen.

### Tafeldienst

So sieht der grundlegende Zyklus zur Laufzeit eines

Blackboard-Systems aus: Blackboard-Einträge werden durch eine Wissensquelle verändert. Dies führt unter Umständen dazu, daß Kontrollinformationen aktualisiert werden (auf dem Blackboard oder separat). Einige Wissensquellen melden einen möglichen Beitrag zum aktuellen Stand der Lösung an. Das Kontrollmodul

wählt anhand von Kontrollinformationen einen Aufmerksamkeitsfokus und eine passende Wissensquelle aus, die zur Anwendung kommt. Möglich ist statt dessen auch eine zweiphasige Selektion, bei der Bedingungs- und Aktionsteile der Wissensquellen getrennt verwaltet und aufgerufen werden. Dieser Zyklus wird so lange

durchlaufen, bis keine Wissensquelle mehr einen Beitrag leisten kann. Dies sollte erst dann der Fall sein, wenn die Lösung auf dem Blackboard steht.

Es gibt eine lange Liste von Bereichen, in denen Blackboard-Systeme erfolgreich eingesetzt wurden. Neben Spracherkennung und Lehr-/Lernsystemen gehören auch Finanz-



## Didaktisches Planen in Lehrsystemen

Blackboard-Architekturen haben inzwischen auch prototypische Anwendungen in computergestützten Lehrsystemen gefunden. Um die Aufgaben in diesem Kontext einzuschätzen, ist es hilfreich, ein wenig weiter auszuholen.

Seit etwa 25 Jahren bemüht man sich, Lernen durch Einsatz von Computern attraktiver und effizienter zu gestalten. Doch lange Zeit wurde von der Flexibilität, die ein Computer ermöglicht, nur wenig ausgenutzt. In einem typischen computerisierten Lehrsystem war der Lehrstoff in eine Reihe von Portionen aufgeteilt, wobei der Übergang von einer Portion auf die nächste etwa vom Wunsch des Benutzers ('Wenn Sie noch etwas über Thema X erfahren wollen, wählen Sie Kapitel Y an...') oder von seinem (oberflächlich geprüften) Wissen abhängig gemacht wurde ('Da Sie mehr als zwei Fragen falsch beantwortet haben, sollten Sie nun Kapitel Z wiederholen...'). Die Portionen selbst enthielten herkömmliche Lehrtexte, möglicherweise von Grafiken oder Animationen unterstützt. Solche 'programmierten' Kurse gibt es auch in Buchform, und in der Tat ist es meist nicht klar, warum man dafür einen Computer braucht. Der motivierende Effekt des neuen Mediums, der bei einem Computernutzen zu registrieren sein mag, verflüchtigt sich spätestens dann, wenn die Augen brennen. Dennoch haben Systeme dieser Art eine gewisse Verbreitung gefunden, etwa als Zugabe zu kommerziellen Programmen wie beispielsweise Textverarbeitungssystemen. Da in solchen Fällen die zu erlernenden Fähigkeiten selbst in der richtigen Bedienung des Computers liegen, ist eine computerisierte Unterweisung besonders hilfreich.

War es das schon? Ja, meint Hubert Dreyfus, amerikanischer Philosoph und prominenter KI-Kritiker. Computer können höchstens dort im Bildungsbereich von Nutzen sein, wo es darum geht, einfach strukturierte Fähigkeiten wie das Lösen bestimmter Rechenaufgaben durch intensiven Drill einzutrainieren, meint Dreyfus. Dennoch sind in den letzten Jahren eine ganze Reihe von Programmsystemen entwickelt worden, die von den beteiligten KI-Wissenschaftlern (mit der ihnen eigenen Unbefangenheit beim Gebrauch solcher Termini) als 'Intelligente Tutorielle Systeme' (ITS) bezeichnet werden.

In einem herkömmlichen Lehrsystem sind im wesentlichen pädagogische Entscheidungen programmiert, in welcher Reihenfolge bestimmte Lehrportionen verabreicht werden sollen. Solche Strategien sind nicht an einen bestimmten Lehrinhalt gebunden, so daß es nahelag, Hilfsprogramme zur Entwicklung beliebiger Lehrsysteme zu vertreiben, die nur noch mit Lehrstoff gefüllt werden müssen. Aber selbst nach dem Füllen wissen solche Systeme eigentlich nichts über das, was sie lehren sollen, weil sie nicht flexibel damit umgehen können. ITS dagegen sind wissensbasiert, das heißt, sie haben Zugriff auf eine Menge Wissen über den Gegenstandsbereich, können dies auf vielfältige Weise einsetzen und daher mehr Verantwortung im Kommunikationsprozeß übernehmen. Daneben erwerben sie aber auch im Dialog mit dem Lernenden Wissen über ihn selbst, wobei sie etwa versuchen, aus seinem Lernverhalten zugrundeliegende falsche oder zutreffende Überzeugungen zu diagnostizieren: Das System lernt also auch, nicht nur der Benutzer. Als dritte Form von Wissen enthalten ITS auch pädagogisches Wissen zur Steuerung des Lehrdialogs.

Das Problem der didaktischen Planung bei Lehrsystemen besteht nun darin, an vielen Punkten der Interaktion Fragen folgender Art (intern) zu beantworten: Soll der Stoff eingeführt/ motiviert/ zusammengefaßt/ zu anderen Themen in Beziehung gesetzt werden/...? Soll das Wissen des Benutzers erhoben werden? Wenn ja, wie? Dabei sind unter anderem die augenblicklichen Ziele, das Wissen über den Benutzer, der Themenbereich und die Kosten zu berücksichtigen. Da alle Komponenten der Entscheidung Veränderungen unterworfen sind, müssen planerische Entscheidungen bei ITS dynamisch gefällt werden.

In der Blackboard-Shell BB1 wurde nun von der kalifornischen FMC Corporation ein Prototyp entwickelt, der solche Planungen durchführen und anpassen kann. Dabei wird ein separates Blackboard für die Kontrollentscheidungen eingesetzt. Das heißt, nicht nur das Lösungs-Blackboard mit Lehrstrategien, Zielen, Benutzermodell und so weiter ist Gegenstand von Veränderungen, sondern auch die Kontrollstrategie selbst kann zur Laufzeit angepaßt werden. (Zur 'seltsamen Schleife' ist es nicht mehr weit.)

analyse, optische Signalverarbeitung, Steuerung autonomer Fahrzeuge, VLSI-Design, Proteinanalyse, Prozeßkontrolle, Krisenmanagement und Produktion dazu. Auf den ersten Blick sind sicherlich kaum Gemeinsamkeiten zwischen diesen Bereichen zu erkennen. Die einzelnen Anwendungen teilen jedoch einige der folgenden Eigenschaften, was sie zu Kandidaten für das Blackboard-Paradigma macht:

- Es sind unterschiedliche Formalismen zur Repräsentation des relevanten Wissens sinnvoll.
- Verschiedenartige Verfahren zur Lösung von Teilproblemen sind zu integrieren.
- Die Probleme erweisen sich als sehr komplex mit einem großen Raum möglicher Lösungen und darüber hinaus als so

schlecht strukturiert, daß sie nicht mit bekannten Verfahren vollständig zu lösen sind. Ein gewisser Grad an Unschärfe und Unsicherheit der relevanten Informationen bleibt vorhanden.

- Eine dynamische Kontrolle der Problemlöseaktivitäten erscheint sinnvoll; a priori lassen sich die Subsysteme nicht zufriedenstellend serialisieren.

### Uni-Spiele

Die starke Unabhängigkeit der Module eines Blackboard-Systems lassen diese Architektur auch für die Parallelverarbeitung geeignet erscheinen.

Eine Reihe von Nachteilen war sicherlich mit dafür verantwortlich, daß Blackboard-Systeme im Gegensatz etwa zu objektorientierten Systemen

immer noch eher ein Fall für universitäre Forschungseinrichtungen als für kommerzielle Entwicklungsabteilungen sind: Sie gelten zunächst als recht langsam, was natürlich ein Preis insbesondere für eine flexible dynamische Kontrollstruktur ist. - die Einführung des Spielers beim Puzzlespiel wird wohl mehr die Qualität als die Geschwindigkeit der Problemlösung positiv beeinflussen. Zuzuschreiben ist die mangelnde Effizienz aber sicherlich auch der geringen Verfügbarkeit entsprechender Entwicklungswerkzeuge, die erst seit kurzem aus den Labors entlassen wurden. Wer sich alles von Hand zusammenstricken muß, erwirbt selten die effektivste Implementierung.

Ein weiteres Hindernis für die rasche Verbreitung mag ge-

wesen sein, daß sich die Vorteile von Blackboard-Systemen nicht auf Anwendungen geringer Komplexität übertragen lassen - wer die Türme von Hanoi, die Geschichte mit den Missionären fressenden Kannibalen oder andere Drosophilae der Programmierdidaktik als Blackboard-System implementiert, könnte leicht die nächste Beförderung verpassen. (ae)

### Literatur zum Weiterlesen

- Robert Englemore & Tony Morgan (Hrsg.), Blackboard Systems, Wokingham, Addison-Wesley 1988.
- V. Jagannathan, Rajendra Doshiawala & Lawrence S. Baum (Hrsg.), Blackboard Architectures and Applications, Boston, Academic Press 1989. 